Аппроксимация. Минимизация функций двух переменных.

I. Аппроксимация (результатов измерений).

И в этот раз начну с Вики: *"Аппроксима́ция, или приближе́ние — научный метод, состоящий в замене одних объектов другими, в том или ином смысле близкими к исходным, но более простыми. Аппроксимация позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов (например, таких, характеристики которых легко вычисляются или свойства которых уже известны)*".

[https://ru.wikipedia.org/wiki/Аппроксимация]

Энциклопедически очень точно и очень общо. Настолько общо, что даже не совсем понятно. Да, аппроксимация (она же приближение) – метод и в самом деле очень общий, применяемый в самых разных формах и в громадном разнообразии приложений. Наши задачи гораздо более узкие, мы будем рассматривать применение аппроксимации, опять цитата оттуда же: " *... для обработки экспериментальных или натурных данных. Тут следует рассматривать два случая: 1) аппроксимирующая функция ограничена диапазоном заданных точек и служит в качестве только интерполирующей зависимости; 2) аппроксимирующая функция выступает в роли физического закона и с её помощью допускается экстраполировать переменные*", причём нас интересует именно второй случай. Первый случай – интерполирование данных – мы рассмотрели в прошлый раз. Чтоб не возвращаться, экстраполяция - это, грубо говоря, распространение резыльтатов наших наблюдений или измерений вне области наблюдений, на неисследованные нами области, совсем уж приблизительно и образно – это попытка предсказать будущее или заглянуть в далёкое и неизвестное прошлое. Да, эти два сюжета – прошлый и сегодняшний – имеют довольно много сходства, по крайней мере, внешнего, но сегодняшняя задачи принципиально отличается от задачи интерполяции. В прошлый раз мы имели результаты экспериментов (измерений ли, вычислений ли, ещё чего-то, неважно) и пытались сделать что-то вроде построения графика, пытались смоделировать проведение плавной линии, соединяющей на графике построенные точки. При этом причины именно такой зависимости величин, их внутренние связи нас не интересовали. Сегодня ситуация совсем иная. У нас как раз есть законы, связывающие наши величины, описывающие результаты экспериментальных данных. Но мы должны определить какие-то параметры этих связей, этих законов. Т.е. результаты экспериментов есть, и это, весьма вероятно, довольно большой массив данных, но объект нашего внимания сегодня именно природа связей этих величин, причины, как именно они связаны, каковы параметры этой связи. Мы понимаем законы (или думаем, что понимаем, предполагаем, возможно у нас есть какие-то гипотезы относительно этих законов), определяющие связи величин, но нам надо найти характеристики этих законов – их параметры. Если в задаче интерполяции мы интерполировали просто данные, точки, произвольные абстрактные точки, то сегодня мы аппроксимируем функцию.

Давайте рассмотрим пример. У нас есть обычный резистор (сопротивление). Мы знаем, что сила тока связана с напряжением законом Ома: $U = R \cdot I$, где U – напряжение, R – сопротивление резистора (внутренним сопротивлением источника напряжения пренебрегаем), I – сила тока в цепи. Закон Ома носит, конечно, эмпирический характер, так что на вопрос "почему?" он ответа не даёт, но нас сейчас интересует не это. Нас сейчас интересует именно вычислить величину сопротивления R. Кажется, что тут такого: подали напряжение, измерили силу тока, разделили, и готово. Но не всё так гладко. Измерения

совершенно естественно сопровождаются какими-то погрешностями, и в результате при одном напряжении мы получим один результат, при другом – другой и т.д. Так что, закон Ома не выполняется? Нет, выполняется конечно, но по одному измерению, выполненному, напомню, с погрешностью (а избежать её полностью в принципе невозможно), определить характеристику этого закона – сопротивление R – невозможно. Среди многих более или менее близких значений R, получаемых каждым отдельным измерением, надо выбрать какое-то одно. Какое? И как это сделать? Вот такого рода вопросы нас и интересуют сегодня.

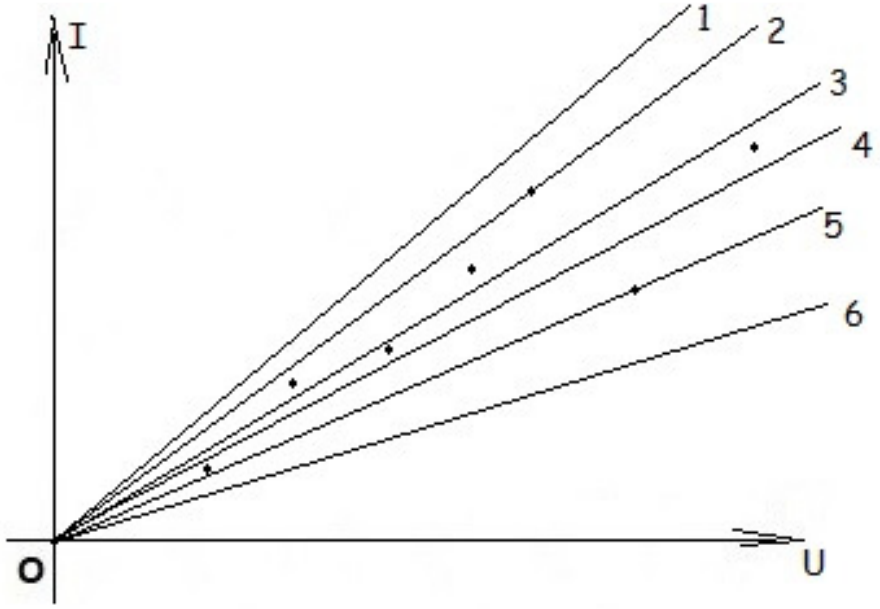


Рисунок 1.

На рисунке 1 для примера изображены семь точек – результатов измерений. И какая из 6 изображённых на рисунке прямых лучше других? Сразу понятно, что первая и шестая прямые явно хуже остальных (а, кстати, хорошо бы как-то выразить словами, чем именно они хуже). Почти также очевидно, что и вторая и пятая прямые тоже нехороши. А вот какая прямая лучше - третья или четвёртая – совсем не очевидно. Кажется на глазок, что третья получше, но кто его знает… И потом, даже если согласиься с тем, что третья лучше (я сказал "если"), то весьма вероятно, что есть и получше неё.

Ответ на вопрос "какое?" почти очевиден. У нас есть выбор из многих возможностей. Какой вариант выбрать? Понятное дело, самый лучший. Ответ, кажется демагогически формальным, но давайте задумаемся чуть-чуть... Вот у нас есть результаты измерений. Вот мы знаем (или предполагаем, что знаем), что результаты измерений связаны, причём эта взаимосвязь описывается как-то более или менее просто. В природе чаще всего так и случается. Да и вообще, простота формулировок, изящность, элегантность выражений очень часто есть свидетельство адекватности наших представлений о мире реальности. Да, я залез в какие-то эстетические категории, с ними всё смутно и плохо формализуемо, но они есть, они реальны, и это трудно оспаривать. Мы рассматриваем зависимости, описываемые какой-то формулой, в нашем примере $U = R \cdot I$. Я умышленно до этого момента избегал использовать слово "формула", чтобы подчеркнуть общность подхода. Теперь я буду только так и говорить, только, пожалуйста, не надо забывать, что описание явления формулами в алгебраической (ну, или, чуть более общо, математической) нотации – есть хоть и очень важный случай, но только частный случай. Итак, есть формула с какими-то не определёнными пока параметрами, есть величины, примерно соответствующие этой формуле. Да-да, соответствующие, но только примерно, есть ещё погрешность измерений.

Перепишем чуть-чуть нашу формулу, вот так: $I = U/R$. И она говорит, что, если подать на резистор некоторое напряжение U_k, то по нему должен проходить ток U_k/R, а между тем измеренное значение силы тока равно I_k, несколько отличающемуся от предсказанной формулой величины. Ага, отличающемуся. Чем больше отличия, тем хуже наша формула соответствует реальности. Так что понятие "самый лучший" в данном случае соответствует малости отличий предсказанных значений от измеренных. И получается, что нам надо найти такое значение R, при котором эти отличия минимальны. Оппа! Так это же готовая задача минимизации, ну, или оптимизации, если хотите. Да! Именно так. Есть отличие значений, полученных по формуле, от измеренных значений. Это отличие зависит от парметров формулы, в данном примере – от одной-единственной величины R, и мы хотим подобрать значение R так, чтобы отличие было минимальным.

Постановка задачи.

Осталось только формализовать понятие *отличия*. Но это как раз, вроде бы, несложно. Давайте оторвёмся от примера с резистором (не навсегда, мы к нему ещё обратимся) и скажем, что у нас есть некоторая функция f, описывающая связь величин. Изменяемую характеристику обозначим, естественно, x, а измеряемую – y. Мы измерили величину y при N значениях x, обозначим их x_1, x_2, x_3, … x_N, и получили, соответственно, N значений y: y_1, y_2, y_3, … y_N. Величины $f(x_k)$ и y_k, как мы уже говорили не совпадают вследствие погрешностей измерений. На сколько? Взглянем на рисунок 2. Точки – это результаты измерений, прямая линия – график функции f. Понятно, что вертикальные линии на рисунке (точнее, длины этих отрезков) – это и есть отличия измеренной величины от величины, предсказанной функцией f.

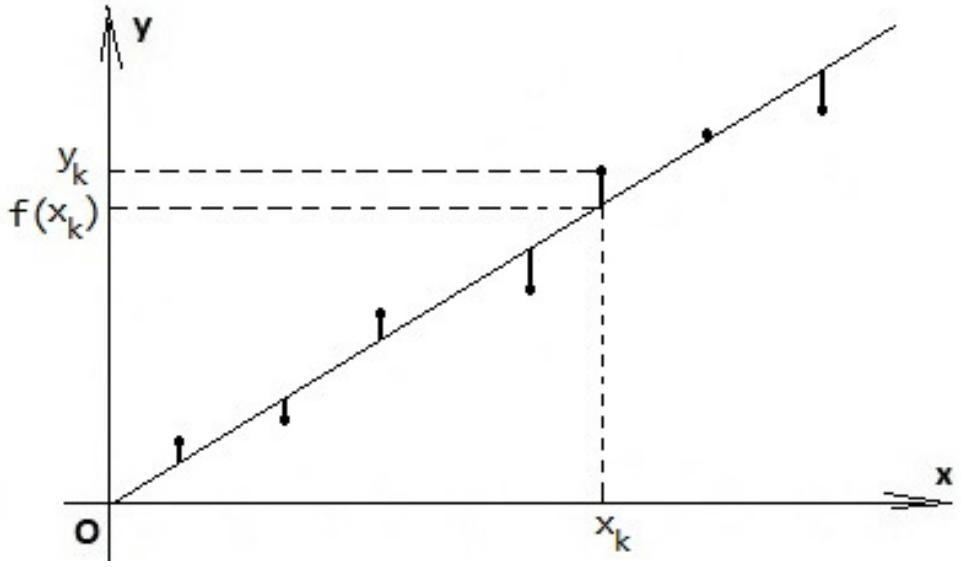


Рисунок 2.

Понятное дело, что при $x=x_k$ отклонение равно $|f(x_k) - y_k|$. Это и есть мера отличия. Но у нас ведь имеется N отличий. Что делать? Нужна какая-то интегрирующая, объединяющая, суммирующая характеристика. Тоже понятно, возьмём сумму отличий (или среднее арифметическое – это одно и то же, только отличаются в N раз, что никак не влияет на поиск параметров функции f). А вот тут немного притормозим. Это далеко не единственный вариант выбора интегрирующей характеристики отличия результатов измерений от значений, вычисляемых в соответствии с функцией f, всё зависит от наших целей. Да-да, помните, в прошлый раз как раз проскакивало такое название – целевая функция (или функция цели). Вскользь замечу – это не функция f, это другая функция, она характеризует функцию f. Целевой функцией мы называли функцию, минимумы/максимумы которой мы искали. И здесь именно это происходит – в зависимости от поставленных целей вид характеристики отличий может изменяться. Приведу несколько обычных и естественных вариантов целевой функции.

$$g_1 = \left|f(x_1) - y_1\right| + \left|f(x_2) - y_2\right| + ... + \left|f(x_N) - y_N\right|$$
$$g_2 = (f(x_1) - y_1)^2 + (f(x_2) - y_2)^2 + ... + (f(x_N) - y_N)^2$$
$$g_\infty = \max\left(\left|f(x_1) - y_1\right|, \left|f(x_2) - y_2\right|, ..., \left|f(x_N) - y_N\right|\right)$$

На вопрос о том, почему такое странное обозначение g_∞, я отвечу только, что и 1, и 2, и ∞ имеют здесь один смысл, что это единая система обозначений, а эти индексы - только её проявления, но что это за система, я умолчу. Это не имеет отношения к нашему сегодняшнему разговору, если не нравится, можете заменить ∞ на 0 или ещё на что-нибудь.

Гораздо интереснее понять смысл этих величин g_1, g_2 и g_∞. Ну, что такое g_1, мы уже обсудили – это суммарное отклонение, что равносильно среднему отклонению. Да, в случае такого выбора целевой функции, мы просто минимизируем среднее отклонение вычисленных величин от измеренных. А g_2 означает почти то же самое. Ну, с качественной точки зрения, какая разница, что брать: модуль или квадрат, лишь бы отклонение было положительным. Да, конечно, результаты эти две целевые функции дают разные, но, как правило, очень близкие друг к другу. А зачем нам вообще g_2? А с ним зачастую проще исполнять выкладки, его проще анализировать, и g_2 довольно часто позволяет получить ответ аналитически, а не численно, просто выразить оптимальные параметры функции f через данные задачи: x_1, x_2, x_3, … x_N и y_1, y_2, y_3, … y_N. Так что выбор целевой функции g_2 встречается довльно часто.

А вот g_∞ имеет совсем другой смысл. В этом случае мы минимизируем наибольшее отклонение. Т.е., если мы тестируем какой-то прибор (в примере с резистором это амперметр и вольтметр), то мы гарантируем, что вот вам параметры функции, которая аппроксимирует заданный набор данных x_1, x_2, x_3, … x_N, y_1, y_2, y_3, … y_N, и мы гарантируем, что ошибка измерений не превосходит вот такой-то величины – минимальное значение g_∞. В первых двух целевых функциях это не так. Вполне возможно, что в одном месте будет дикое отклонение вычисленной величины от измеренной, зато во всех других будет всё тютелька в тютельку. Ну, почти... Это может быть следствием какого-то выброса, ну, например стол с приборами в момент измерения тряхнули или ещё чего. g_∞ к таким вещам очень чувствительна. Зато она гарантирует, что нигде ошибка не будет слишком большой.

Так что какой вариант выбрать – зависит от обстоятельств, от наших целей. Возможны и другие целевые функции в задаче аппроксимации, но не будем о них. В подавляющем большинстве случаев применяется одна из трёх перечисленных, либо их "взвешенные" варианты. Это означает вот что

$$g_1^w = w_1 \cdot \left|f(x_1) - y_1\right| + w_1 \cdot \left|f(x_2) - y_2\right| + ... + w_1 \cdot \left|f(x_N) - y_N\right|$$
$$g_2^w = w_1 \cdot (f(x_1) - y_1)^2 + w_1 \cdot (f(x_2) - y_2)^2 + ... + w_1 \cdot (f(x_N) - y_N)^2$$
$$g_\infty^w = \max\left(w_1 \cdot \left|f(x_1) - y_1\right|, w_1 \cdot \left|f(x_2) - y_2\right|, ..., w_1 \cdot \left|f(x_N) - y_N\right|\right)$$

Здесь w_1, w_2, w_3, … w_N – некоторые заранее заданные положительные коэффициенты, весовые коэффициенты их называют или просто веса. Смысл тоже довольно прозрачен: чем больше весовой коэффициент w_k, тем сильнее влияние отличия $f(x_k)$ от y_k на целевую функцию. Т.е. если w_k большое, то параметры функции f будут подбираться так, что отклонение $f(x_k)$ от y_k было по возможности меньше; если же w_k маленькое, то мы довольно

толерантны к отклонению $f(x_k)$ от y_k, да пусть будет большим, всё равно оно внесёт небольшой вклад в суммарную погрешность.

Решаем пример с резистором.

Но давайте посмотрим на пример с резистором. Изменяемая величина там – напряжение U, измеряемая – сила тока I. Заменим U_k на x_k, I_k – на y_k, а величину 1/R обозначим буквой a. И тогда закон Ома запишется в виде $y = a \cdot x$, и соответствующие целевые функции будут выглядеть так:

$$g_1(a) = |a \cdot x_1 - y_1| + |a \cdot x_2 - y_2| + \ldots + |a \cdot x_N - y_N|$$

$$g_2(a) = (a \cdot x_1 - y_1)^2 + (a \cdot x_2 - y_2)^2 + \ldots + (a \cdot x_N - y_N)^2$$

$$g_\infty(a) = \max\left(|a \cdot x_1 - y_1|, |a \cdot x_2 - y_2|, \ldots, |a \cdot x_N - y_N|\right)$$

Я написал все три функции, но это не означает, что надо минимизировать все три сразу. Нет, каждая из них даёт свою наилучшую аппроксимацию, и какую из них нам брать – это вопрос к постановщику задачи.

Обратите внимание на очень важный момент: у всех трёх функций появился аргумент. И на этом месте тоже хорошо бы потоптаться. Хотя здесь ситуация подобна аналогичной ситуации в задаче интерполяции: иксы и игреки x_1, x_2, x_3, … x_N, y_1, y_2, y_3, … y_N – это входные данные задачи, а величина a – переменная, и мы ищем такое её значение, при котором функция g (любая из трёх или ещё какая) дотигает минимального значения. Да, да, a – есть аргумент функции g. На самом деле всё естественно, если чуть-чуть подумать. g – характеристика аппроксимирующей функции – функции f, т.е. величина g полностью определяется параметрами функции f. А в нашем примере функция f определяется одним параметром a, т.е. g – есть функция одной переменной, и эта переменная - a.

И теперь мы моментально можем применить любую из написанных нами (правда ведь, написанных?) процедур минимизации – то ли половинное деление, то ли “золотое сечение”, то ли Фибоначчи.

Именно так, если функция f задаётся одним параметром, то целевая функция g (которая характеризует отклонение) зависит от одного этого параметра. И мы можем применить к ней любую из процедур численной минимизации функций с одной переменной.

Есть, конечно, ещё мелкий (в данном случае) вопрос об определении отрезка отделения при поиске минимума, но он как раз в данном случае решается совсем просто: вычисляем для всех пар x_k, y_k соответствующее значение параметра функции f, и тогда левый край отрезка отделения – минимальное значение параметра, правый край – максимальное. Так, для задачи про резистор левый край – это минимальное из всех y_k/x_k, правый край – максимальное.

В качестве сильно необязательного дополнения я приведу поиск оптимального значения параметра a для целевой функции g_2.

$$g_2(a) = (a \cdot x_1 - y_1)^2 + (a \cdot x_2 - y_2)^2 + \ldots + (a \cdot x_N - y_N)^2 =$$
$$= (x_1^2 + x_2^2 + \ldots + x_N^2) \cdot a^2 - 2(x_1 y_1 + x_2 y_2 + \ldots + x_N y_N) \cdot a + (y_1^2 + y_2^2 + \ldots + y_N^2)$$

Получили квадратичную функцию с переменной a. Коэффициент при a^2 положителен (сумма квадратов потому что), значит у этой квадратичной функции есть минимум, который достигается при

$$a = \frac{x_1 y_1 + x_2 y_2 + \ldots + x_N y_N}{x_1^2 + x_2^2 + \ldots + x_N^2}$$

Это и есть наилучшее значение параметра a, значение, при котором функция $f(x)=a \cdot x$ наилучшим образом аппроксимирует набор данных $x_1, x_2, x_3, \ldots x_N, y_1, y_2, y_3, \ldots y_N$. Конечно, наилучшим в смысле целевой функции g_2.

Разбирать это на занятии можно, но совсем не обязательно. Тем не менее, результат можно использовать для контроля правильности написанной программы. Ну, а если для g_2 всё в порядке, то, скорее всего, и для g_1, и для g_∞ всё будет хорошо.

Аппроксимация однопараметрической функции.

Эта задача у нас уже решена. Здесь я просто сформулирую вместе результаты и выводы.

Задача состояла в следующем. Есть какие-то величины, связь которых описывается функцией с одним параметром – это функция f и параметр a. Есть набор измерений этих величин, причём вследствие погрешности измерений они не совсем точно удовлетворяют этой зависимости – это величины $x_1, x_2, x_3, \ldots x_N, y_1, y_2, y_3, \ldots y_N$. Задача состоит в том, чтобы подобрать значение параметра a, с которым функция f аппроксимирует (приближает, соответствует) измеренные величины наилучшим образом.

Решение. Выбираем целевую функцию, характеризующую качество аппроксимации (точнее размер погрешности – чем больше значение целевой функции, тем качество хуже) – это функция g. В качестве g можно выбрать, например, функцию g_1, или g_2, или g_∞. Функция g зависит от a и только от a. Ищем точку минимума функции g, применяя какую-нибудь процедуру численной минимизации, и получаем искомое значение параметра a.

Аппроксимация функции с двумя параметрами.

А что делать, если функция определяется двумя параметрами?
Вот, например, рассмотрим такую ситуацию: вертикально вверх вылетает камень со скоростью v_0, сопротивление воздуха считаем пренебрежимо малым. Тогда высота камня h спустя время t от начала полёта определяется соотношением

$$h = v_0 t - g t^2/2,$$

где g – ускорение свободного падения.
У нас имеются замеры высоты камня $h_1, h_2, h_3, \ldots h_N$ в N различных моментов времени $t_1, t_2, t_3, \ldots t_N$, соответственно. При этом зависимость высоты h от времени полёта t включает в себя два параметра – v_0 и g.

Что же делать? Да ничего особенного - делаем всё то, что уже описано выше: для каждого , t_k вычисляем $f(t_k)$ и находим отличие вычисленной величины от измеренного значения h_k. Затем собираем из них какую-нибудь величину, объединяющую все отличия вместе, например g_1, или g_2, или g_∞ и ищем её минимальное значение.

Отличие в точке t_k равно $|f(t_k) - h_k| = |v_0 t_k – g t_k^2/2 - h_k|$ и мы получаем такие выражения для функций g_1, g_2 и g_∞:

$$g_1(v_0, g) = \left|v_0 \cdot t_1 - g \cdot t_1^2/2 - h_1\right| + \left|v_0 \cdot t_2 - g \cdot t_2^2/2 - h_2\right| + \ldots + \left|v_0 \cdot t_N - g \cdot t_N^2/2 - h_N\right|$$

$$g_2(v_0, g) = (v_0 \cdot t_1 - g \cdot t_1^2/2 - h_1)^2 + (v_0 \cdot t_2 - g \cdot t_2^2/2 - h_2)^2 + \ldots + (v_0 \cdot t_N - g \cdot t_N^2/2 - h_N)^2$$

$$g_\infty(v_0, g) = \max\left(\left|v_0 \cdot t_1 - g \cdot t_1^2/2 - h_1\right|, \left|v_0 \cdot t_2 - g \cdot t_2^2/2 - h_2\right|, \ldots, \left|v_0 \cdot t_N - g \cdot t_N^2/2 - h_N\right|\right)$$

И остаётся только для любой (каждой или какой-нибудь – неважно) из этих функций найти минимум. Но вот тут-то мы и упираемся. Мы умеем минимизировать функцию одной переменной, а здесь у нас их две – два параметра функции f - v_0 и g. А это уже другая история. Вот ею и займёмся. И когда мы с ней разберёмся, останется применить полученный алгоритм поиска минимума функции двух переменных к функциям g_1 или g_2 или g_∞.

II. Функции двух переменных . Минимизация функций двух переменных.

Мы поговорим о том, что из себя представляют функции двух переменных и построим один алгоритм поиска минимума для них. Вообще-то существует много разных алгоритмов поиска минимума функции двух переменных (и не только двух, а и нескольких). Это не означает, что бери любой, и будет тебе счастье. Как правило, и в данном случае именно так обстоят дела, наличие большого количества алгоритмов решения какой-то задачи означает, что задача пока ещё далека от окончательного решения, а, возможно, что его и вовсе не существует. Не будем углубляться в такие уже философские материи. Просто построим один алгоритм, не самый эффективный, но достаточно эффективный. А вообще эта история живая, развивающаяся вовсю. И очень интересная. Но я остановлюсь на этом, и просто скажу, что в материалах к занятию я выкладываю совершенно, на мой взгляд, замечательную статью

З.Тьмеладзе. Нелинейное программирование. Журнал "Квант", №1, 1976; стр. 28-34.

В ней излагается обзор различных подходов и методов поиска минимума функции двух переменных, при этом в ней нет ни одной формулы вообще. А статья, повторю, интереснейшая. Читается легко, написана очень понятно, всё очень физично и геометрично, даже если эта тема не особо цепляет, всё равно прочитать её – чистое удовольствие.

Визуализация функции двух переменных.

Действительно, а как нам изобразить "график" функции с двумя переменными f(x, y)? Для функции одной переменной всё понятно, а как быть с двумя переменными? Попробуем поступать аналогично. Берём какие-то значения переменных x и y, находим соответствующую точку. Стоп. Где находим? На плоскости, естественно, их же две. Ну, да, ничего больше и не остаётся. И теперь двигаемся от этой точки вертикально на f(x, y), вверх, если f(x, y)>0 или вниз, если f(x, y)<0. Вертикально, т.е. вдоль оси Oz, все наши построения мы исполняем не на листе бумаги, не на плоскости, а в пространстве. Отмечаем полученную точку с координатами (x, y, f(x,y)) и повторяем процесс ещё для каких-то точек - чем больше, тем лучше. И теперь соединяем построенные точки плавной... Чем плавной? Поверхностью плавной. Так что график (уже без всяких там кавычек) функции двух переменных есть какая-то поверхность в пространстве. Хорошо, что функция двух переменных, и у нас хватает размерностей, чтобы это всё держать в голове, вот было бы три переменных, пришлось бы представлять мысленно 4-мерное пространство. А это довольно непривычно ☺. Я приведу пару примеров графиков функций двух переменных, просто для понимания.

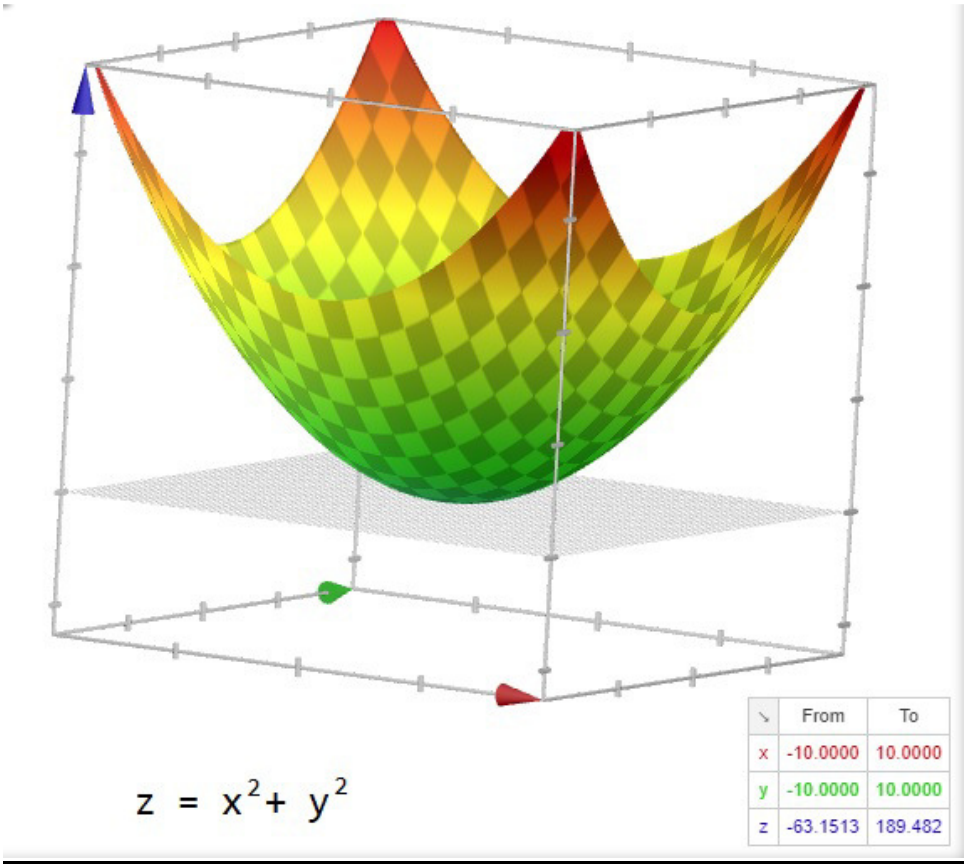


Рисунок 3. График функции $z = x^2 + y^2$

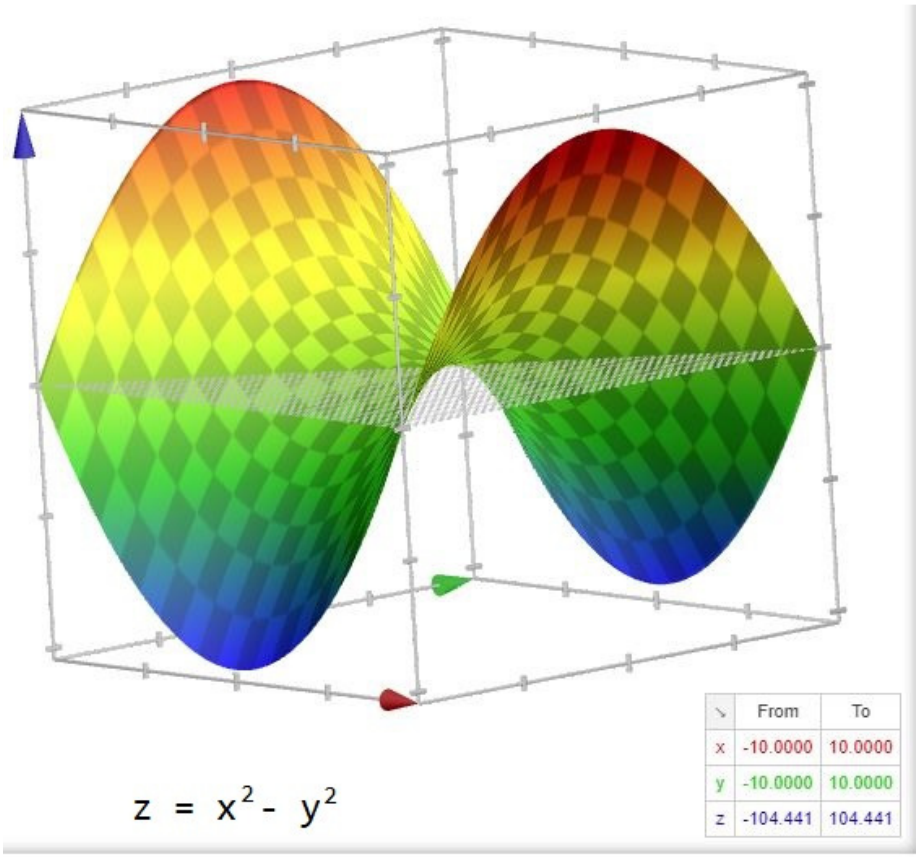


Рисунок 4. График функции $z = x^2 - y^2$

Оба рисунка построены с помощью google. Просто забиваете в поиск x^2+y^2 (или x^2-y^2, понятно, что можно и другие функции) и получаете рисунок, который ещё можно крутить, чтобы рассматривать поверхность с разных сторон. Выделенная плоскость на

рисунках – это плоскость z=0, та самая плоскость, на которой мы откладываем x и y, аналог оси Ox для построения привычных графиков функций одной переменной. Клеточки на поверхностях – это образы единичных клеточек в плоскости xOy, то, во что клеточки на плоскости переходят.

Есть и другой способ изображения визуализации функции двух переменных – изолинии, линии на плоскости xOy, соответствующие одинаковым значениям f(x,y). Звучит, может быть непривычно, но все их видели. На географических картах часто именно изолиниями изображают рельефы (в таком случае изолинии называются изогипсами) и моря-океаны (линии одинаковой глубины называются изобатами). Возможно кто-то помнит, что в метеопрогнозах часто показывают карты с линиями одинакового атмосферного давления (изобарами). Приведу для примера рисунок, на котором показано построение изолиний некоторой функции:

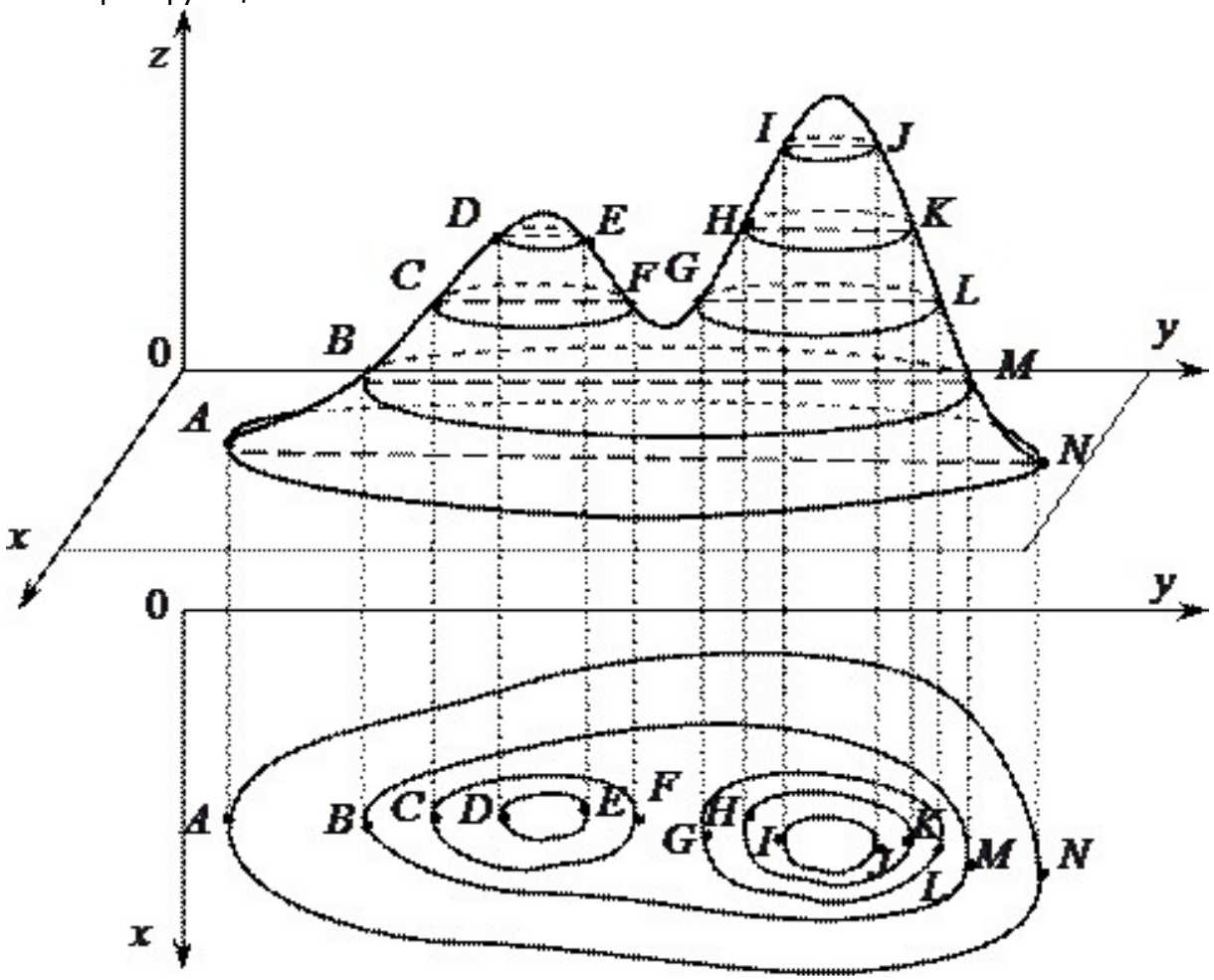


Рисунок 5. Построение изолиний.

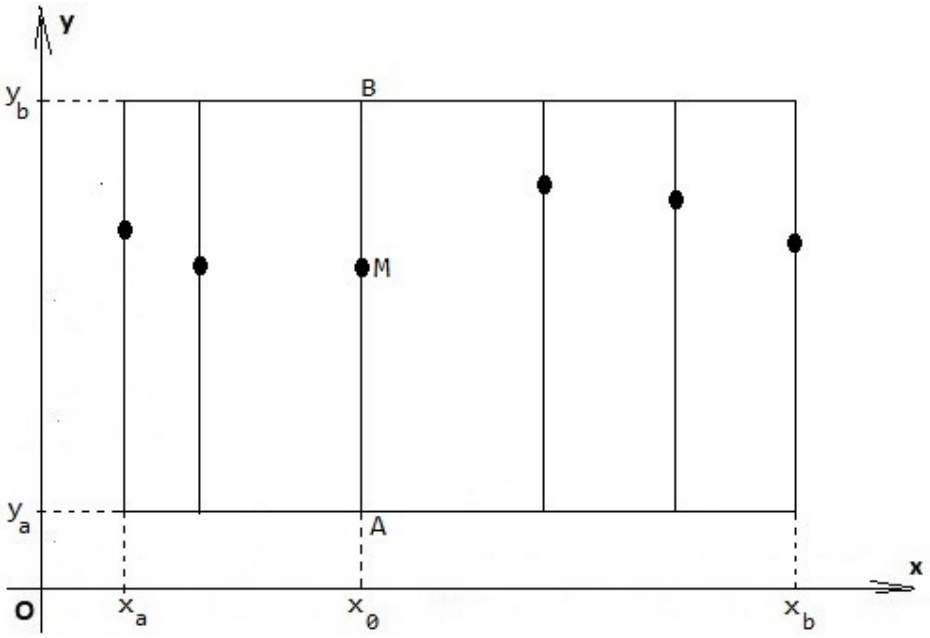
Верхняя часть рисунка – график некоторой функции, нижняя – та же функция, изображённая изолиниями.

Минимизация функции двух переменных.

Кажется, это самый непростой вопрос этого занятия, хотя он довльно простой и ясный, если въехать. Но надо это всё компактно уложить в голове, и тогда всё становится действительно простым до очевидности. Для этого надо приложить некоторые усилия, но это обычное дело - укладывание в голове требует усилий. Да и мы с таким уже сталкивались не раз.

Метод, повторю, довольно естественный – сведём задачу поиска минимума функции двух переменных к задаче поиска минимума функции одной переменной. Разумеется, за такие свдения к более простой задаче надо чем-то платить. И плата здесь будет такая, что мы будем искать минимум функции одной переменной многократно, но всё-таки умеренно много, да мы и не особо будем сегодня заботиться об эффективности, в разумных пределах, конечно.

Собственно, главные слова уже сказаны – сводим двумерную задачу к одномерной. Причём довольно простым способом.



Итак, ищем минимум некоторой функции f(x, y). Взглянем на рисунок 4. Там выделен прямоугольник $x_a \le x \le x_b$, $y_a \le y \le y_b$. Это та область, в которой мы ищем минимум. На рисунке изображена только плоскость xOy, подобно тому, как мы изображали только ось Ox, когда строили методы одномерной минимизации (одномерной – имеется в виду минимизация функции одной переменной). Нас интересуют только передвижения по аргументам функции, а значения функции мы, да, будем сравнивать, но это и всё, они нам на рисунке не очень-то и нужны.

Давайте зафиксируем какой-нибудь x из интервала $[x_a, x_b]$, обозначим его x_0. Пробежимся по отрезку от точки $A(x_0, y_a)$ до точки $B(x_0, y_b)$, вычисляя в каждой точке f(x, y). Мысленно, конечно, ведь точек бесконечно много. Что мы получаем в итоге? Для всех точек $x = x_0$, т.е. x не изменяется, изменяется при этом только y. Иначе говоря, мы вычисляли значения некоторой функции от y, назовём её q: $q_{x0}(y) = f(x_0, y)$. $q_{x0}(y)$ есть функция одной переменной y (x_0 в ней играет роль параметра и, повторю, не изменяется). Мы можем найти точку минимума функции $q_{x0}(y)$ – на рисунке она выделена и обозначена M – ведь мы умеем находить точку минимума одномерной функции. Точка M зависит только от x_0 (ну, и от функции f, конечно), т.е. величина $f(M) = f(x_M, y_M) = f(x_0, y_M)$ полностью определяется величиной x_0. f(M) – есть наименьшее значение функции f при $x = x_0$, и полностью определяется величиной x_0, т.е. $f(M) = h(x_0)$, где h – некоторая одномерная функция, функция переменной x: h(x) – это минимальное значение функции f(x, y) среди всех y от y_a до y_b. Одномерная функция! Ну, так и вычислим её минимум. Понятно, что это и есть минимум функции f на всём прямоугольнике. Всё!

Давайте посмотрим, что же у нас получилось. Мы свели вычисление минимума двумерной функции f к вычислению минимума одномерной функции h. При этом вычисление одного значения функции h требует решения задачи одномерной минимизации, т.е. задачу

одномерной минимизации нам надо решать многократно. Вот вам и ответ на вопрос, зачем нам было выжимать эффективность из методов одномерной минимизации, зачем мы вращались с методами Фибоначчи и/или "золотого сечения".

Для большей ясности я приведу программу на Go, реализующую этот алгоритм:

```
package main

import "fmt"

type (
     Function1D func(float64) float64
     Function2D func(float64, float64) float64
)

func minimum2D(left, right, bottom, top float64, F2 Function2D)
(float64, float64) {
     h:= (func (x float64) float64 {
          q:= (func (y float64) float64  {
               return F2(x, y)
          } )
          ymin:= minimum1D(bottom, top, q)
          return F2(x, ymin)
     }  )

     xmin:= minimum1D(left, right, h)
     ymin:= minimum1D(bottom, top, ( func (y float64) float64 {
                                        return f(xmin, y)
                                   }  )  )
     return xmin, ymin
}

func minimum1D(start, finish float64, F1 Function1D) float64 {
     // Any procedure of the 1D minimization
     var pmin float64    //minimum point
     // ...
     return pmin
}

func f(x, y float64) float64 {
     var res float64
     // ...
     return res
}

func main() {
     var left, right, bottom, top float64
     // left, right, bottom, top = ...
     fmt.Println(minimum2D(left, right, bottom, top,  f))
}
```

Примечание к программе. В этой программе появляется, во-первых, уже известное детям, но пока ещё мало пользованное понятие функционального типа и переменных функционального типа – на него надо бы обратить внимание. А ещё, кажется впервые, появляются анонимные функции (аж три раза, там где: h:=… , q:=… и ymin:=…), но с ними как раз всё довольно естественно, так что обратить внимание надо всенепременнейше и

слегка потоптаться тоже, но именно слегка, заморачиваться на этом не надо. И понятно всё более или менее, и это не в последний раз мы с анонимными функциями сталкиваемся. *Конец примечания.*

Конечно, это не совсем программа, это заготовка, в которую надо включить какую-нибудь процедуру одномерной минимизации, а также определить функцию f и прямоугольник, на котором мы отделили её минимум: left и right – это x_a и x_b на рисунке 4, bottom и top – это y_a и y_b.

Остаётся вопрос о том, какими свойствами должен обладать прямоугольник отделения минимума функции f – тот самый прямоугольник, который на рисунке 4. Не будем это подробно обсуждать, скажем только что в этом прямоугольнике функция f должна быть дважды унимодальной (это означает, что если мы фиксируем x – так, как мы это сделали – то получающаяся одномерная функция q(y) должна быть унимодальной для каждого x; и то же самое должно происходить, если мы фиксируем y и получаем одномерную функцию, зависящую от x). А ещё скажем, что это условие не страшное, и для функций g_1, g_2 и g_∞ (напомню, если вы забыли – давно это было – что это те самые функции, которые мы используем для оценки отличия аппроксимирующей функции от измеренных данных) выполняется практически всегда, если только точка минимума попадает в прямоугольник. А точка минимума – это и есть искомые параметры аппроксимирующей функции, их мы, как правило, приблизительно, пусть даже очень грубо, можем оценить из физического (или какого-нибудь ещё) смысла задачи.

Задачи и упражнения.

I. Однопараметрическая аппроксимация.

"Резистор"

Даны результаты измерений силы тока I в амперах для восьми различных значений напряжения U в вольтах:

Напряжение, U (В)	10,0	20,0	30,0	40,0	50,0	60,0	70,0	80,0
Сила тока, I (А)	0,34	0,68	1,01	1,33	1,66	2,02	2,28	2,62

Напряжение и сила тока подчиняются закону Ома: $I = a \cdot U$, где $a = 1/R$, R – сопротивление резистора, a – проводимость резистора.

Определить проводимость резистора для трёх целевых функций

$$g_1(a) = |a \cdot U_1 - I_1| + |a \cdot U_2 - I_2| + \ldots + |a \cdot U_N - I_N|$$

$$g_2(a) = (a \cdot U_1 - I_1)^2 + (a \cdot U_2 - I_2)^2 + \ldots + (a \cdot U_N - I_N)^2$$

$$g_\infty(a) = \max\left(|a \cdot U_1 - I_1|, |a \cdot U_2 - I_2|, \ldots, |a \cdot U_N - I_N|\right)$$

Входные данные находятся в файле `01.resistor.dat`.

"Ускорение свободного падения". Для определения ускорения свободного падения g ($м/с^2$) измерены расстояния, которые пролетел свободно падающий груз:

Время, t (сек.)	0.2	0.4	0.6	0.8	1.0
Расстояние, h (м)	0.1960	0.7850	1.7665	3.1405	4.9075

Найти g для трёх целевых функций g_1, g_2 и g_∞, исходя из соотношения

$$h = gt^2/2.$$

Извините, но ускорение свободного падения традиционно обозначается той же буквой g, что я выбрал для целевых функций, но, надеюсь, это не приведёт к путанице.

Входные данные находятся в файле `01.gravitation.dat`.

II. Двупараметрическая аппроксимация.

"Камень". Вертикально вверх вылетает камень со скоростью v_0, сопротивление воздуха считаем пренебрежимо малым. Высота камня h спустя время t от начала полёта определяется соотношением

$$h = v_0 t - gt^2/2,$$

где g – ускорение свободного падения.

В таблице приведены результаты измерений местоположения (высоты) камня в зависимости от времени :

Время t, сек	Высота h, м	Время t, сек	Высота h, м	Время t, сек	Высота h, м
0.0	0.00	1.6	20.53	3.2	16.00
0.2	3.99	1.8	21.35	3.4	13.67
0.4	7.51	2.0	21.79	3.6	10.98
0.6	10.65	2.2	21.84	3.8	7.82
0.8	13.41	2.4	21.40	4.0	4.31
1.0	15.81	2.6	20.66	4.2	0.42
1.2	17.73	2.8	19.49		
1.4	19.33	3.0	17.96		

Найти параметры v_0 и g, аппроксимируя данную зависимость.

Входные данные находятся в файле `02.stone.dat`.

"Река". В таблице приведены характеристики основных притоков некоторой реки:

Длина L, км	Водосбор s, км^2	Длина L, км	Водосбор s, км^2
498	25100	202	4440
100	1680	937	98200
93	1220	99	1250
226	4000	121	1800
170	1940	76	520
140	2060	115	1345
67	791	105	1220
98	1140	325	6990

Найти формулу (параметры a и b) вида $s(L) = a \cdot L^b$, аппроксимирующую данную зависимость.

Входные данные находятся в файле `02.river.dat`.